Deletion on Trajectories and Commutative Closure*

Michael Domaratzki[†] Alexandru Mateescu[‡] Kai Salomaa[§]
Sheng Yu[¶]

Abstract

We study the recently introduced deletion on trajectories operation in the case when the set of trajectories is associative and complete. We define a congruence relation associated with a language L and a set of trajectories T and the corresponding factor monoid is called the syntactic (L, T)-monoid. We prove that the syntactic monoid is finite if and only if the commutative closure of L is regular.

1 Introduction

Recently there has been considerable interest in deletion and scattered deletion operations, see e.g. Ito et al. [5, 6] and the references listed there. Scattered deletion can be viewed as the inverse of shuffle. On the other hand, shuffle as well as many other basic language operations are obtained as special cases of shuffle on trajectories considered by Mateescu et al. [10, 11]. Domaratzki [2] has introduced a similar unifying framework for deletion-like operations.

A set of trajectories is complete if the shuffle defined by it admits some composition of words of any length. A set of trajectories is said to be associative if the corresponding shuffle operation is associative. We consider algebraic properties of deletion on complete and associative sets of trajectories. First we establish a number of basic properties related to deletion on trajectories and the commutative closure of languages. We define a congruence relation associated with a language L on the monoid determined by the shuffle operation along a complete and associative set of trajectories. We show that the congruence is decidable for given context-free languages when the commutative closure of L is effectively regular and T

*Work supported by NSERC grants OGP0041630 and OGP0147224 and an NSERC PGS-B graduate scholarship.

[†]School of Computing, Queen's University, Kingston, Ontario K7L 3N6, Canada. Email: domaratz@cs.queensu.ca

[‡]University of Bucharest, Faculty of Mathematics, Academiei, 14, Bucharest 1, Romania. Email: alema@correu.urv.es

[§]School of Computing, Queen's University, Kingston, Ontario K7L 3N6, Canada. Email: ksalomaa@cs.queensu.ca

[¶]Department of Computer Science, University of Western Ontario, London, Ontario N6A 5B7, Canada. Email: syu@csd.uwo.ca

satisfies certain assumptions. The congruence relation induces a syntactic monoid associated with the language L and set of trajectories T. As the main result we extend for deletion along trajectories the characterization given by Mateescu [9] for the case of unrestricted scattered deletion that relates the finiteness of the syntactic monoid with regularity of the commutative closure of the given language.

The cardinality of a syntactic monoid associated with a set of trajectories can be finite, countably infinite or uncountable. We establish closure properties of the class of languages having a finite syntactic monoid with respect to any complete and associative set of trajectories. We consider the question whether for all sets of trajectories the syntactic monoid may be countably infinite or uncountable.

2 Definitions

Below we fix Σ to be an arbitrary finite alphabet. Let $V = \{1, 2\}$. A *trajectory* is a word $t \in V^*$.

The shuffle along trajectories was originally defined in [10]. Let $\alpha = ax$, $\beta = by$, $a, b \in \Sigma$, $x, y \in \Sigma^*$. The shuffle of α with β on the trajectory ct, where $c \in V$ and $t \in V^*$ is defined as follows:

$$ax \sqcup\!\sqcup_{ct} by = \begin{cases} a(x \sqcup\!\sqcup_t by), & \text{if } c = 1, \\ b(ax \sqcup\!\sqcup_t y), & \text{if } c = 2. \end{cases}$$

If $\alpha = ax$ and $\beta = \lambda$, then

$$ax \sqcup\!\sqcup_{ct} \lambda = \begin{cases} a(x \sqcup\!\sqcup_t \lambda), & \text{if } c = 1, \\ \emptyset, & \text{otherwise.} \end{cases}$$

If $\alpha = \lambda$ and $\beta = by$, then

$$\lambda \sqcup\!\sqcup_{ct} by = \begin{cases} b(\lambda \sqcup\!\sqcup_t y), & \text{if } c = 2, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Finally we set $\lambda \sqcup\!\sqcup_\lambda \lambda = \lambda$ and $\alpha \sqcup\!\sqcup_\lambda \beta = \emptyset$ always when $\alpha \neq \lambda$ or $\beta \neq \lambda$.

The operation $\sqcup\!\sqcup_t$ is extended to languages and sets of trajectories in the natural way. If L_1, L_2 are languages and T is a set of trajectories, then

$$L_1 \sqcup\!\sqcup_T L_2 = \bigcup_{\alpha \in L_1, \beta \in L_2, t \in T} \alpha \sqcup\!\sqcup_t \beta.$$

When $T = V^*$, $\sqcup\!\sqcup_T$ reduces to the unrestricted shuffle operation which is denoted simply by $\sqcup\!\sqcup$.

Deletion on trajectories has been introduced in [2]. According to the definition of [2], a trajectory defining a deletion operation is a string over $\{i, d\}$. The below definition uses 1 in place of i and 2 in place of d. Let $\alpha = ax$, $\beta = by$, $a, b \in \Sigma$, $x, y \in \Sigma^*$. If $t \in V^*$ is a trajectory, then the *t-deletion* of β from α is defined as follows:

$$ax \rightsquigarrow_{ct} by = \begin{cases} a(x \rightsquigarrow_t by), & \text{if } c = 1, \\ x \rightsquigarrow_t y, & \text{if } c = 2 \text{ and } a = b \\ \emptyset, & \text{otherwise.} \end{cases}$$

If $\alpha = ax$ and $\beta = \lambda$, then

$$ax \rightsquigarrow_{ct} \lambda = \begin{cases} a(x \rightsquigarrow_t \lambda), & \text{if } c = 1, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Also, $\lambda \rightsquigarrow_t \beta = \lambda$ if $t = \beta = \lambda$ and $\lambda \rightsquigarrow_t \beta = \emptyset$ otherwise.

For example, $ababa \rightsquigarrow_{21121} ab = baa$ and $ababa \rightsquigarrow_{11221} ab = aba$.

The t-deletion operation is extended to languages and sets of trajectories. If L_1, L_2 are languages and T is a set of trajectories, then

$$L_1 \rightsquigarrow_T L_2 = \bigcup_{\alpha \in L_1, \beta \in L_2, t \in T} (\alpha \rightsquigarrow_t \beta).$$

If α is a word, then the *commutative closure* of α is $\mathrm{com}(\alpha) = \{\beta \mid \forall a \in \Sigma,\ |\alpha|_a = |\beta|_a\}$, where $|\gamma|_a$ denotes the number of occurrences of the symbol a in the word γ. If L is a language, then

$$\mathrm{com}(L) = \bigcup_{\alpha \in L} \mathrm{com}(\alpha).$$

Definition 2.1 We say that a set of trajectories $T \subseteq \{1, 2\}^*$ is *complete* if for all $m, n \in \mathbb{N}$ there exists $t \in T$ such that $|t|_1 = m$ and $|t|_2 = n$.

The following proposition is obvious.

Proposition 2.1 *Let L_1, L_2 be languages and T is a set of trajectories.*

1. $L_1 \subseteq com(L_1)$.
2. *If* $L_1 \subseteq L_2$, *then* $com(L_1) \subseteq com(L_2)$.
3. $com(L_1 \cup L_2) = com(L_1) \cup com(L_2)$.
4. $com(com(L_1)) = com(L_1)$.
5. *If T is complete, then* $com(L_1 \sqcup\!\sqcup_T L_2) = com(L_1 L_2)$.
6. *If* $T = com(T)$, *then* $com(L_1 \sqcup\!\sqcup_T L_2) = com(L_1) \sqcup\!\sqcup_T com(L_2)$.
7. *If T is complete, then* $com(L_1 \sqcup\!\sqcup_T L_2) = com(com(L_1)com(L_2))$.
8. $com(L_1 \sqcup\!\sqcup_T L_2) = com(L_1) \sqcup\!\sqcup_{com(T)} com(L_2)$.

As a consequence of Proposition 2.1 (8) we have:

Corollary 2.1 1. *If T is complete then* $com(L_1 \sqcup\!\sqcup_T L_2) = com(L_1) \sqcup\!\sqcup com(L_2)$.

2. *If T_1 and T_2 are complete, then* $com(L_1 \sqcup\!\sqcup_{T_1} L_2) = com(L_1 \sqcup\!\sqcup_{T_2} L_2)$.

Also the following proposition is easy to prove.

Proposition 2.2 *For any languages L_1 and L_2 and for any set T of trajectories:*

$$L_1 \rightsquigarrow_T L_2 = \{w \in \Sigma^* \mid \exists v \in L_2 : (w \sqcup\!\sqcup_T v) \cap L_1 \neq \emptyset\}$$

As a consequence of Proposition 2.2 we get the following:

Corollary 2.2 *Let T_1 and T_2 be complete sets of trajectories. For any languages L and K we have*

$$com(L) \rightsquigarrow_{T_1} K = com(L) \rightsquigarrow_{T_2} K.$$

Corollary 2.3 *Let T be a complete set of trajectories and L is some language. Let K, J be languages such that $com(K) = com(J)$. Then*

$$com(L) \rightsquigarrow_T K = com(L) \rightsquigarrow_T J.$$

Actually the statements of Corollaries 2.2 and 2.3 could be combined by having two distinct complete sets of trajectories in Corollary 2.3 but we have stated the results in the simpler form that will be used later.

Remark 2.1 *From now on we will consider only complete sets of trajectories T such that the operation $\sqcup\!\sqcup_T$ is associative.*

A set of trajectories is said to be associative if the operation $\sqcup\!\sqcup_T$ is associative. An exact characterization of associative sets of trajectories is given in [10]. We do not repeat the somewhat technical conditions here and instead refer the reader to Proposition 4.7 of [10]. From the characterization it follows that for a regular language T it is decidable whether or not T is associative, and in [10] it is shown also that the language family consisting of associative sets of trajectories is an anti-AFL. Examples of non-associative sets of trajectories include the set of trajectories $(12)^*$ defining balanced literal shuffle and the set $1^*2^*1^*$ defining insertion.

By the above assumption we know that the construct $\mathcal{L}_T = (\mathcal{P}(\Sigma^*), \sqcup\!\sqcup_T, \lambda)$ is a monoid.

Definition 2.2 Let L be a fixed language and let T be a fixed set of trajectories. The relation induced by L and T in $\mathcal{L}_T$, denoted $\sim_{L,T}$, is defined as follows. For any $L_1, L_2 \subseteq \Sigma^*$,

$$L_1 \sim_{L,T} L_2 \text{ if and only if } \mathrm{com}(L) \rightsquigarrow_T L_1 = \mathrm{com}(L) \rightsquigarrow_T L_2.$$

Proposition 2.3 *The relation $\sim_{L,T}$ is a congruence on $\mathcal{L}_T$.*

Proof. Note that $\sim_{L,T}$ is an equivalence relation. Now assume that $L_1 \sim_{L,T} L_2$ and let A be a language. We will prove that $A \sqcup\!\sqcup_T L_1 \sim_{L,T} A \sqcup\!\sqcup_T L_2$, that is,

$$\mathrm{com}(L) \rightsquigarrow_T (A \sqcup\!\sqcup_T L_1) = \mathrm{com}(L) \rightsquigarrow_T (A \sqcup\!\sqcup_T L_2).$$

Let u be in $\mathrm{com}(L) \rightsquigarrow_T (A \sqcup\!\sqcup_T L_1)$. Using Proposition 2.2 there exists $z \in A \sqcup\!\sqcup_T L_1$ such that $(u \sqcup\!\sqcup_T z) \cap \mathrm{com}(L) \neq \emptyset$.

Thus there are $x \in A$ and $y \in L_1$ such that $z \in x \sqcup\!\sqcup_T y$ and $(u \sqcup\!\sqcup_T z) \cap \mathrm{com}(L) \neq \emptyset$. Hence $u \sqcup\!\sqcup_T (x \sqcup\!\sqcup_T y)) \cap \mathrm{com}(L) \neq \emptyset$. Since T is associative, it follows that $(u \sqcup\!\sqcup_T x) \sqcup\!\sqcup_T y \cap \mathrm{com}(L) \neq \emptyset$.

Therefore, there is some $\alpha_1 \in (u \sqcup\!\sqcup_T x) \sqcup\!\sqcup_T y$ such that $\alpha_1 \in \mathrm{com}(L)$. Let $\beta_1 \in u \sqcup\!\sqcup_T x$ be such that $\alpha_1 \in \beta_1 \sqcup\!\sqcup_T y$. Thus, $(u \sqcup\!\sqcup_T x) \cap (\alpha_1 \rightsquigarrow_T y) \neq \emptyset$, as β_1 is in the intersection.

As $(u \sqcup\!\sqcup_T x)$ has a non-empty intersection with $com(L) \rightsquigarrow_T L_1$, it also has a non-empty intersection with $com(L) \rightsquigarrow_T L_2$, since they coincide. Let $\alpha_2 \in com(L)$ and $y' \in L_2$ be such that $(u \sqcup\!\sqcup_T x) \cap (\alpha_2 \rightsquigarrow_T y') \neq \emptyset$. Thus, $\alpha_2 \in (u \sqcup\!\sqcup_T x) \sqcup\!\sqcup_T y' = u \sqcup\!\sqcup_T (x \sqcup\!\sqcup_T y')$. As $x \in A$ and $y' \in L_2$, we have that there is some $z' \in x \sqcup\!\sqcup_T y' \subseteq A \sqcup\!\sqcup_T L_2$ such that $u \sqcup\!\sqcup_T z' \cap \mathrm{com}(L) \neq \emptyset$, as α_2 is in this intersection. Thus,

$$u \in \mathrm{com}(L) \rightsquigarrow_T (A \sqcup\!\sqcup_T L_2),$$

and we have shown that $\mathrm{com}(L) \rightsquigarrow_T (A \sqcup\!\sqcup_T L_1) \subseteq \mathrm{com}(L) \rightsquigarrow_T (A \sqcup\!\sqcup_T L_2)$. The converse inclusion follows by symmetry.

Also, a similar proof shows that $L_1 \sqcup\!\sqcup_T B \sim_{L,T} L_2 \sqcup\!\sqcup_T B$ where B is any language. Hence $\sim_{L,T}$ is a congruence. ■

Definition 2.3 If L is a fixed language and T is a fixed set of trajectories, then the factor monoid $\mathcal{L}_T / \sim_{L,T}$ is called the *syntactic (L,T)-monoid,* denoted $\mathcal{L}(L,T)$.

Definition 2.4 A language L over Σ is *commutatively saturated* by an equivalence relation $\sim$ on $\mathcal{P}(\Sigma^*)$ if and only if, for any languages L_1 and L_2 over Σ,

$$\text{if } L_1 \cap \mathrm{com}(L) \neq \emptyset \text{ and } L_1 \sim L_2, \text{ then } L_2 \cap \mathrm{com}(L) \neq \emptyset.$$

Proposition 2.4 *1. $com(L)$ is commutatively saturated by $\sim_{L,T}$.*

2. $\sim_{L,T}$ is the largest congruence on $\mathcal{L}_T$ with the above property.

Proof. 1. Assume that $L_1 \sim_{L,T} L_2$ and that $L_1 \cap \mathrm{com}(L) \neq \emptyset$. There exists $w \in \mathrm{com}(L) \cap L_1$ and thus $w \rightsquigarrow_T w = \lambda$. Hence $\lambda \in (\mathrm{com}(L) \rightsquigarrow_T L_1)$ and consequently $\lambda \in (\mathrm{com}(L) \rightsquigarrow_T L_2)$.

Now there exist $u \in \mathrm{com}(L)$ and $u \in L_2$ such that $u \rightsquigarrow_T u = \lambda$. Hence $u \in \mathrm{com}(L) \cap L_2$ and thus $\mathrm{com}(L) \cap L_2 \neq \emptyset$.

2. Let $\approx$ be a congruence of $\mathcal{L}_T$ such that L is commutatively saturated by $\approx$. Assume that $L_1 \approx L_2$ and consider the language $\mathrm{com}(L) \rightsquigarrow_T L_1$.

Let x be in $\mathrm{com}(L) \rightsquigarrow_T L_1$. Note that $\mathrm{com}(L) \cap (x \sqcup\!\sqcup_T L_1) \neq \emptyset$, $(x \sqcup\!\sqcup_T L_1) \approx (x \sqcup\!\sqcup_T L_2)$ and hence $\mathrm{com}(L) \cap (x \sqcup\!\sqcup_T L_2) \neq \emptyset$. Therefore there exists $w \in L_2$ such that $(x \sqcup\!\sqcup_T w) \cap \mathrm{com}(L) \neq \emptyset$. We can conclude that $x \in \mathrm{com}(L) \rightsquigarrow_T L_2$. Hence $\mathrm{com}(L) \rightsquigarrow_T L_1 \subseteq \mathrm{com}(L) \rightsquigarrow_T L_2$. The converse inclusion is symmetric.

We have shown that $L_1 \sim_{L,T} L_2$ and thus $\approx \subseteq \sim_{L,T}$. ■

To conclude this section we investigate the decidability of the relation $\sim_{L,T}$. We say that a set T is *letter-bounded* if there exist $b_1, b_2, \ldots, b_k \in \{1,2\}$ such that $T \subseteq b_1^* b_2^* \cdots b_k^*$. The following result has been established in [3]:

Proposition 2.5 *Let $T \subseteq \{1,2\}^*$ be a letter-bounded regular set of trajectories. Then for any regular language R and any context-free language X, the language $R \rightsquigarrow_T X$ is effectively regular.*

When R is regular and T is regular and letter-bounded, the language $R \rightsquigarrow_T X$ is, in fact, regular for any language X but, in order to make the result effective, we need to assume that X is context-free [3]. As a consequence of the above we get immediately:

Theorem 2.1 *Let $T \subseteq \{1,2\}^*$ be a letter-bounded regular set of trajectories. Let L be such that $com(L)$ is effectively regular. Then given context-free languages L_1, L_2, it is decidable whether or not $L_1 \sim_{L,T} L_2$.*

Deletion operations in the literature modelled by letter-bounded trajectories include, for example, (sequential) deletion, left- and right-quotient, bi-polar deletion [7] and k-deletion [8]. The proof of Proposition 2.5 relies on the letter-boundedness property and we do not know whether the decidability result can be extended for general regular sets of trajectories.

3 Characterization result

Below we give a characterization of languages L such that the syntactic (L,T)-monoid is finite for any complete and associative set of trajectories T. For our proof we will use the following technical property.

Lemma 3.1 *Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be a minimal DFA such that $L(\mathcal{A})$ is commutatively closed (that is, $com(L(\mathcal{A})) = L(\mathcal{A})$.) Then for any $Y \subseteq Q$ and $K \subseteq \Sigma^*$: $\delta(Y, K) = \delta(Y, com(K))$.*

Proof. The proof is given in [9] as part of the proof of Theorem 3.1 of that paper. ■

We introduce the following notation. Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be a DFA. The monoid $\mathcal{M}(\mathcal{A}) = (\mathcal{P}(Q)^{\mathcal{P}(Q)}, \cdot, 1_{\mathcal{P}(Q)})$ is called the monoid of *hyperactions* of $\mathcal{A}$.

Theorem 3.1 *Let L be a language over Σ and T a complete and associative set of trajectories. The following conditions are equivalent:*

(i) $com(L)$ is a regular language.

(ii) There exists a congruence $\approx$ on $\mathcal{L}_T$ of finite index such that L is commutatively saturated with respect to $\approx$.

(iii) The syntactic (L,T)-monoid $\mathcal{L}(L,T)$ is finite.

Proof. First we show that (i) implies (ii). Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be the minimal DFA such that $\mathrm{com}(L) = L(\mathcal{A})$. Let $\varphi : \mathcal{L}_T \to \mathcal{M}(\mathcal{A})$ be the function such that for any $K \in \mathcal{L}_T$, $\varphi(K) : X \to X$ is the function defined by the condition

$$\varphi(K)(Y) = \delta(Y, K), \quad (Y \subseteq Q)$$

We show that φ is a morphism. For any $J, K \subseteq \Sigma^*$ and for any $Y \subseteq Q$, the following equalities hold.

$$\begin{aligned}
\varphi(J \sqcup\!\sqcup_T K)(Y) &= \delta(Y, J \sqcup\!\sqcup_T K) = \delta(Y, \mathrm{com}(J \sqcup\!\sqcup_T K)) = \delta(Y, \mathrm{com}(JK)) \\
&= \delta(Y, JK) = \delta(\delta(Y, J), K) = \delta(\varphi(J)(Y), K) \\
&= \varphi(K)(\varphi(J)(Y)) = (\varphi(K) \circ \varphi(J))(Y).
\end{aligned}$$

Here the fourth equality uses Lemma 3.1. Since Y is arbitrary we conclude that $\varphi(J \sqcup\!\sqcup_T K) = \varphi(J) \cdot \varphi(K)$ for any J and K. Hence φ is a morphism of monoids.

The kernel of φ, $\mathrm{Ker}\varphi = \{(A, B) \mid \varphi(A) = \varphi(B)\}$, is a congruence of $\mathcal{L}_T$. Because $\mathcal{M}(\mathcal{A})$ is finite, it follows that $\mathrm{Ker}\varphi$ has finite index (that is, has finitely many classes).

We show that L is commutatively saturated by $\mathrm{Ker}\varphi$. Assume that L_1 and L_2 are languages such that $\varphi(L_1) = \varphi(L_2)$ and assume that $L_1 \cap \mathrm{com}(L) \neq \emptyset$. Hence $\delta(q_0, L_1) \cap F \neq \emptyset$. But $\varphi(L_1)(q_0) = \varphi(L_2)(q_0)$ and thus $\delta(q_0, L_1) = \delta(q_0, L_2)$. Therefore $\delta(q_0, L_2) \cap F \neq \emptyset$ and thus $L_2 \cap \mathrm{com}(L) \neq \emptyset$.

Second we show that (ii) implies (iii). By Proposition 2.4 (2), it follows that $\mathrm{Ker}\varphi \subseteq \sim_{L,T}$ and hence

$$\mathrm{card}(\mathcal{L}_T/\mathrm{Ker}\varphi) \geq \mathrm{card}(\mathcal{L}_T/\sim_{L,T}).$$

We know that $\mathrm{card}(\mathcal{L}_T/\mathrm{Ker}\varphi)$ is finite and, consequently, $\mathcal{L}(L, T)$ is a finite monoid.

Finally we show that (iii) implies (i). Consider the DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ that is defined as follows. We set $Q = \mathcal{P}(\Sigma^*)/\approx$. Note that Q is a finite set. The $\approx$-class of a language K is denoted by $[K]$. We choose $q_0 = [\lambda]$ and $F = \{[K] \mid K \cap \mathrm{com}(L) \neq \emptyset\}$. The transition function $\delta : Q \times \Sigma \to Q$ is given by $\delta([B], \sigma) = [B \sqcup\!\sqcup_T \sigma]$. Note that δ is well defined, i.e., if $B \approx B'$, then $B \sqcup\!\sqcup_T \sigma \approx B' \sqcup\!\sqcup_T \sigma$. Now we prove that $L(\mathcal{A}) = \mathrm{com}(L)$.

For the inclusion from left to right, consider $w \in L(\mathcal{A})$. This means that $\delta([\lambda], w) \in F$. If $w = w_1 w_2 \cdots w_n$, where $w_i \in \Sigma$, $1 \leq i \leq n$, then it is easy to see that $\delta([\lambda], w) = [w_1 \sqcup\!\sqcup_T w_2 \sqcup\!\sqcup_T \cdots \sqcup\!\sqcup_T w_n]$. Hence there is a language K such that $K \cap \mathrm{com}(L) \neq \emptyset$ and $K \approx w_1 \sqcup\!\sqcup_T \cdots \sqcup\!\sqcup_T w_n$. But L is commutatively saturated by $\approx$ and thus $(w_1 \sqcup\!\sqcup_T \cdots \sqcup\!\sqcup_T w_n) \cap \mathrm{com}(L) \neq \emptyset$. Therefore, $w_1 \sqcup\!\sqcup_T \cdots \sqcup\!\sqcup_T w_n$ is a commutation of a word from L, and $w = w_1 \cdots w_n \in \mathrm{com}(L)$. Hence $L(\mathcal{A}) \subseteq \mathrm{com}(L)$.

For the converse inclusion consider arbitrary $w \in \mathrm{com}(L)$. Hence there exists $u \in L$ such that $w \in \mathrm{com}(u)$. Now if $w = w_1 w_2 \cdots w_n$, $w_i \in \Sigma$, $1 \leq i \leq n$, then $\delta([\lambda], w) = [w_1 \sqcup\!\sqcup_T \cdots \sqcup\!\sqcup_T w_n]$. Since $w_1 \sqcup\!\sqcup_T \cdots \sqcup\!\sqcup_T w_n \subseteq \mathrm{com}(L)$ by the definition of the automaton $\mathcal{A}$ it follows that $[w_1 \sqcup\!\sqcup_T \cdots \sqcup\!\sqcup_T w_n]$ is in the set F. ■

We may note that for trace languages [1] there exist regularity characterizations that are similar to the result of Theorem 3.1 in the particular case when, on the one hand, we consider the trace monoid corresponding to the total commutation relation and, on the other hand, the set of trajectories is chosen to be $T = \{1, 2\}^*$ (representing scattered deletion [6]). Such a characterization result is given in Proposition 4.1 of [1]. However, in this case the syntactic congruence of a trace language (as defined on p. 479 of [1]) is not exactly the same as the congruence $\sim_{L,T}$ considered here.

Next we present a result that gives a method to compute the syntactic (L, T)-monoid. First we recall the definition of the transition monoid associated with a DFA.

Definition 3.1 Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be a DFA. Any subset K of Σ^* defines in a natural way a function $\hat{K} : \mathcal{P}(Q) \to \mathcal{P}(Q)$ defined by setting $\hat{K}(Y) = \delta(Y, K)$ for any $Y \subseteq Q$. The submonoid of $\mathcal{M}(\mathcal{A})$ generated by the functions $\hat{K}$, $K \subseteq \Sigma^*$, is called the monoid of *hypertransitions* and it is denoted $\mathcal{T}(\mathcal{A})$.

Theorem 3.2 *Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be a minimal DFA such that $L(\mathcal{A}) = com(L)$. For any (complete and associative) set of trajectories T, the monoid $\mathcal{L}(L, T)$ is isomorphic with a submonoid of $\mathcal{T}(\mathcal{A})$.*

Proof. We define the function $\varphi : \mathcal{T}(\mathcal{A}) \to \mathcal{L}(L, T)$ by setting $\varphi(\hat{K}) = [K]$ where $[K]$ is the $\sim_{L,T}$-class of $K \subseteq \Sigma^*$.

We show that φ is well defined. Consider $K, J \subseteq \Sigma^*$ such that $\hat{K} = \hat{J}$. Thus for all $Y \subseteq Q$, $\delta(Y, K) = \delta(Y, J)$. In particular, $\delta(q_0, K) = \delta(q_0, J)$ implies that

$$\mathrm{com}(L) \rightsquigarrow_U K = \mathrm{com}(L) \rightsquigarrow_U J \tag{1}$$

where $U = 2^*1^*$ is the trajectory defining left-quotient. Now (1) and Corollary 2.2 imply that $\mathrm{com}(L) \rightsquigarrow_T K = \mathrm{com}(L) \rightsquigarrow_T J$, that is, $[K] = [J]$.

Next we show that φ is a morphism. Let K, J be any subsets of Σ^*. Then

$$\varphi(\hat{K} \cdot \hat{J}) = \varphi(\widehat{KJ}) = [KJ], \text{ and, } \varphi(\hat{K}) \cdot \varphi(\hat{J}) = [K](\sqcup\!\sqcup_T / \sim_{L,T})[J] = [K \sqcup\!\sqcup_T J].$$

Since T is complete, by Proposition 2.1 (5), $\mathrm{com}(KJ) = \mathrm{com}(K \sqcup\!\sqcup_T J)$ and thus Corollary 2.3 gives us $[KJ] = [K \sqcup\!\sqcup_T J]$. Now the claim follows since, by definition, φ is surjective. ∎

The proof of Theorem 3.2 gives a method to compute the syntactic (L, T)-monoid. This is illustrated in the following example.

Example 3.1 Consider the non-context-free language

$$L = \{ww \mid w \in \{a, b\}^*\} \cdot \{\lambda, b\}.$$

(The factor $\{\lambda, b\}$ is included because for the language $L_0 = \{ww \mid w \in \{a, b\}^*\}$ the monoid $\mathcal{L}(L_0, T)$ would be fairly large, having 16 elements, as the reader can verify.)

The commutative closure of L, $\mathrm{com}(L) = \{w \in \{a, b\}^* \mid |w|_a \equiv 2 \pmod 2\}$, is regular and it is accepted by the minimal DFA $\mathcal{A} = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{q_0\})$ where $\delta(q_0, a) = \delta(q_1, b) = q_1$ and $\delta(q_1, a) = \delta(q_0, b) = q_0$.

The monoid of hypertransitions is generated by subsets of $\{\lambda, a\}$ with identities $a^2 \equiv \lambda$, $b \equiv \lambda$. The table of hypertransitions is

	$\emptyset$	λ	a	$\{\lambda, a\}$
q_0	$\emptyset$	q_0	q_1	$\{q_0, q_1\}$
q_1	$\emptyset$	q_1	q_0	$\{q_0, q_1\}$

According to the construction in the proof of Theorem 3.2, for any (associative and complete) set of trajectories T the syntactic (L,T)-monoid consists of four elements $[\emptyset]$, $[\lambda]$, $[a]$, $[\{\lambda, a\}]$ and the multiplication table of $\mathcal{L}(L,T)$ is

$\sqcup\!\sqcup_T / \sim_{L,T}$	$[\emptyset]$	$[\lambda]$	$[a]$	$[\{\lambda, a\}]$
$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$	$[\emptyset]$
$[\lambda]$	$[\emptyset]$	$[\lambda]$	$[a]$	$[\{\lambda, a\}]$
$[a]$	$[\emptyset]$	$[a]$	$[\lambda]$	$[\{\lambda, a\}]$
$[\{\lambda, a\}]$	$[\emptyset]$	$[\{\lambda, a\}]$	$[\{\lambda, a\}]$	$[\{\lambda, a\}]$

4 Closure properties

Since $\mathcal{P}(\Sigma^*)$ is uncountable, the cardinality of a syntactic (L,T)-monoid $\mathcal{L}(L,T)$ may be finite, countably infinite ($\aleph_0$) or uncountable ($\aleph_1$). Theorem 3.1 provides a characterization of cases where the monoid $\mathcal{L}(L,T)$ is finite. We consider the following language families.

Definition 4.1 The family of languages with a finite syntactic (L,T)-monoid is denoted $\mathcal{F}in(T)$, i.e.,

$$\mathcal{F}in(T) = \{L \mid \mathcal{L}(L,T) \text{ is finite } \}.$$

Analogously we define

$$\mathcal{F}_0(T) = \{L \mid \mathrm{card}(\mathcal{L}(L,T)) = \aleph_0\},$$
$$\mathcal{F}_1(T) = \{L \mid \mathrm{card}(\mathcal{L}(L,T)) = \aleph_1\}.$$

Note that by Theorem 3.1 for any set of trajectories T, the monoid $\mathcal{L}(L,T)$ is finite if and only if $\mathrm{com}(L)$ is regular and hence the class $\mathcal{F}in(T)$ does not depend on the set of trajectories T, and we use for it simply the notation $\mathcal{F}in$. (According to Remark 2.1 we restrict consideration to complete and associative sets of trajectories.)

Since the commutative closure of every finite language is regular, we observe that all finite languages are in $\mathcal{F}in$.

Theorem 4.1 *The family $\mathcal{F}in$ is not comparable with any family in the Chomsky hierarchy.*

Proof. Above we have observed that $\mathcal{F}in = \mathcal{F}in(T)$ is independent of the set of trajectories T. Hence the result follows from [9] where it is established for the case of unrestricted shuffle given by $T = \{1,2\}^*$. ■

Theorem 4.2 *For a given context-free language L it is decidable whether or not L is in $\mathcal{F}in$.*

Proof. For any context-free language L we can effectively construct a letter-equivalent regular language R. Obviously $\mathrm{com}(L) = \mathrm{com}(R)$ and hence $\mathcal{L}(L,T) = \mathcal{L}(R,T)$. It is decidable whether or not, for a given regular language R, $\mathrm{com}(R)$ is regular [4]. The claim follows from the fact that $L \in \mathcal{F}in$ if and only if $\mathrm{com}(R)$ is regular. ■

The results of Theorems 4.3 and 4.4 below follow from the corresponding closure and non-closure results in [9] using the observation that $\mathcal{F}in = \mathcal{F}in(T)$ is independent of the set of trajectories T.

Theorem 4.3 *The family $\mathcal{F}in$ is closed under the following operations: union, mirror image, commutative closure and cycle, where* $\mathrm{cycle}(L) = \{uv \mid vu \in L\}$.

Theorem 4.4 *The family $\mathcal{F}in$ is not closed under complement, intersection, intersection with regular languages, Kleene closure, morphisms or inverse moprhisms.*

Also, it turns out that $\mathcal{F}in$ is not closed under deletion on trajectories, not even when the set of trajectories is in the family $\mathcal{F}in$.

Theorem 4.5 *There exist $L_1, L_2, T \in \mathcal{F}in$ such that $L_1 \rightsquigarrow_T L_2$ is not in $\mathcal{F}in$.*

Proof. Consider the languages $L_1 = \{ca^nb^n \mid n \geq 0\} \cup b^*a^*c$, $L_2 = \{c\}$ and $T = 21^*$. Now L_1, L_2 and T are in $\mathcal{F}in$ since $\mathrm{com}(L_1) = \{a,b\}^*c\{a,b\}^*$, $\mathrm{com}(L_2) = L_2$ and $\mathrm{com}(T) = 1^*21^*$. However, $L_1 \rightsquigarrow_T L_2 = \{a^nb^n \mid n \geq 0\}$ and $\mathrm{com}(L_1 \rightsquigarrow_T L_2) = \{w \mid |w|_a = |w|_b\}$ is non-regular. ■

For closure under shuffle along trajectories we have the following characterization.

Theorem 4.6 *Let T be a trajectory. Then T is in $\mathcal{F}in$ if and only if $\mathcal{F}in$ is closed under shuffle along T.*

Proof. For the "only if" part consider $L_1, L_2, T \in \mathcal{F}in$. Since $\mathrm{com}(L_1)$, $\mathrm{com}(L_2)$, $\mathrm{com}(T)$ are all regular, it follows that $\mathrm{com}(L_1) \sqcup\!\sqcup_{\mathrm{com}(T)} \mathrm{com}(L_2)$ is regular [10]. By Proposition 2.1 (8) it follows that $\mathrm{com}(L_1 \sqcup\!\sqcup_T L_2)$ is regular and thus $L_1 \sqcup\!\sqcup_T L_2 \in \mathcal{F}in$.

For the converse implication assume that $\mathcal{F}in$ is closed under shuffle along T. Since $\mathrm{com}(0^*) = 0^*$ is regular we have $0^* \in \mathcal{F}in$, and similarly we observe that $1^* \in \mathcal{F}in$. Thus

$$T = 0^* \sqcup\!\sqcup_T 1^* \ \in \mathcal{F}in.$$

■

There exist languages L such that for any set of trajectories T the syntactic (L,T)-monoid is uncountable.

Theorem 4.7 *For any complete set of trajectories T, $\mathcal{F}_1(T) \neq \emptyset$.*

Proof. Let $\Sigma = \{a,b\}$ and $L = \{w \mid \ |w|_a = |w|_b\}$. Let T be an arbitrary complete set of trajectories. We will show that $\mathrm{card}(\mathcal{L}(L,T)) = \aleph_1$.

Define $f : \mathcal{P}(a^*) \to \mathcal{L}(L,T)$ by setting $f(A) = [A]$, $A \subseteq a^*$. In order to prove the claim it is sufficient to show that f is injective.

Let $A_1, A_2 \subseteq a^*$ be such that $A_1 \neq A_2$. By symmetry we can assume that $A_1 - A_2 \neq \emptyset$ and choose $a^k \in A_1 - A_2$. Since $a^k \notin A_2$ it follows that $b^k \notin (\mathrm{com}(L) \rightsquigarrow_{L,T} A_2)$. On the other hand, since $a^k \in A_1$, T is complete, and in words of $\mathrm{com}(L)$ we can "re-arrange" the symbols in any order it follows that $b^k \in (\mathrm{com}(L) \rightsquigarrow_{L,T} A_1)$. Hence $[A_1] \neq [A_2]$ and f is injective. ■

It is shown in [9] that $\mathcal{F}_0(T) \neq \emptyset$ for $T = \{1,2\}^*$. We do not know whether the construction giving a countably infinite syntactic (L,T)-monoid can be extended for any set of trajectories T and it remains an open question whether $\mathcal{F}_0(T) \neq \emptyset$ for all complete and associative sets of trajectories T.

References

[1] V. Diekert and Y. Métivier, Partial commutation and traces. In: *Handbook of Formal Languages, Vol. 3* (G. Rozenberg and A. Salomaa, Eds.), Springer-Verlag, 1997, pp. 457–533.

[2] M. Domaratzki, Deletion along trajectories. School of Computing, Queen's University, Technical report 2003–464.

[3] M. Domaratzki and K. Salomaa, Results on Decidability of Trajectory-Based Equations. Manuscript in preparation, 2003.

[4] S. Ginsburg and E.H. Spanier, Bounded regular sets. *Proc. Amer. Math. Soc.* (1966) 1043–1049.

[5] M. Ito, L. Kari and G. Thierrin, Insertion and deletion closure of languages. *Theoretical Computer Science* **183** (1997) 3–19.

[6] M. Ito, L. Kari and G. Thierrin, Shuffle and scattered deletion closure of languages. *Theoretical Computer Science* **245** (2000) 115–133.

[7] L. Kari, On language equations with invertible operations. *Theoretical Computer Science* **132** (1994) 129–150.

[8] L. Kari and G. Thierrin, k-insertion and k-deletion closure of languages. *Soochow J. of Mathematics* **21** (1995) 479–495.

[9] A. Mateescu, Scattered deletion and commutativity. *Theoretical Computer Science* **125** (1994) 371–371.

[10] A. Mateescu, G. Rozenberg and A. Salomaa, Shuffle on trajectories: Syntactic constraints. *Theoretical Computer Science* **197** (1998) 1–56.

[11] A. Mateescu, K. Salomaa and S. Yu, On fairness of many-dimensional trajectories. *J. Automata, Languages and Combinatorics* **5** (2000) 145–157.